

## Working With Streams

In this lab we will see how streams can be used for maintaining application data or also the contents generated from application into the stream. Following are the streams offered by .NET Framework under System.IO namespace:

- Stream -> Abstract class.
- FileStream
- MemoryStream.
- BinaryStream.

Data can be read and write using StreamReader, StreamWriter, BinaryReader etc. In this lab we will explore FileStream, IsolatedStorage and File Encoding mechanism.

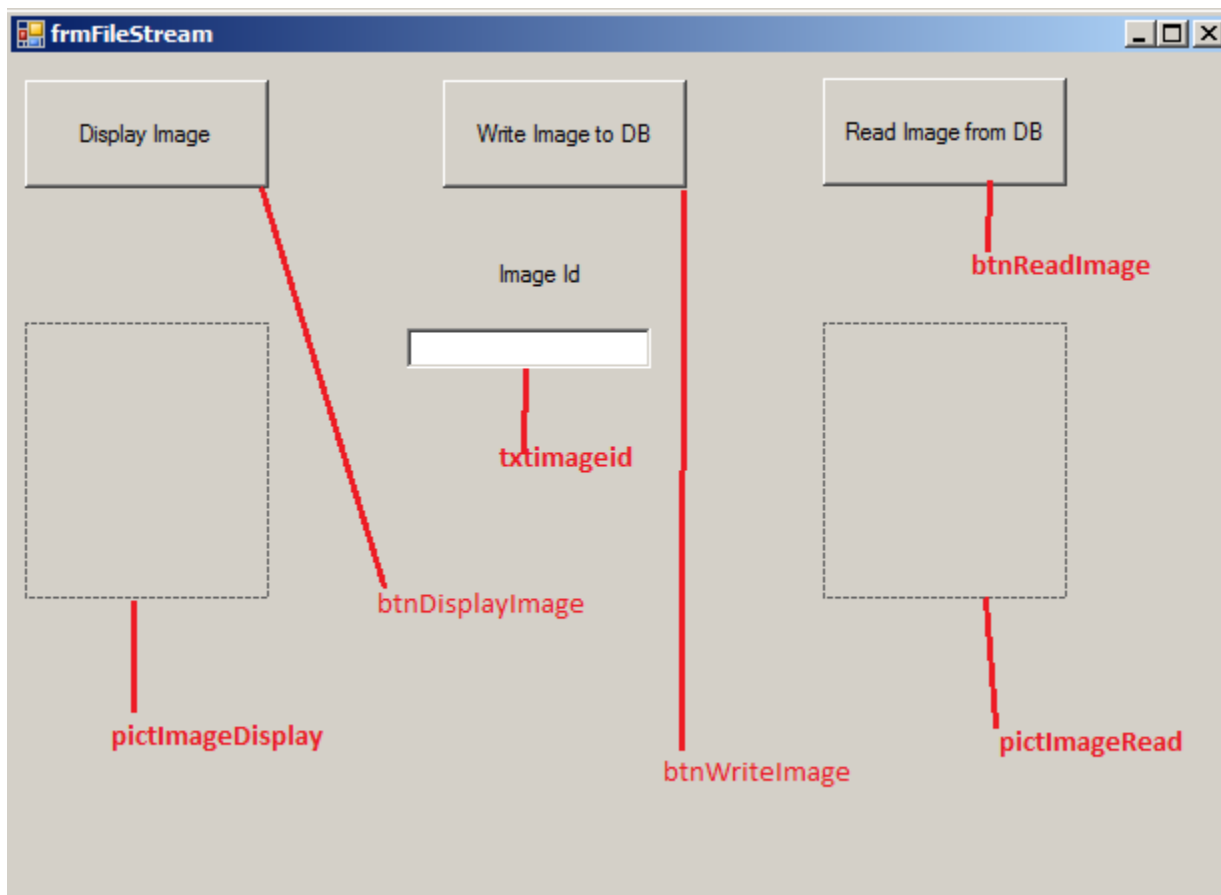
### Exercise 1: Using FileStream to storage Image into database.

**Task 1:** Open Sql Server and create the below table of name ImageTable:

WIN-UPTAHM9H...o.ImageTable			
	Column Name	Data Type	Allow Nulls
PK	ImageId	int	<input type="checkbox"/>
	Image	image	<input type="checkbox"/>
			<input type="checkbox"/>

**Task 2:** Open VS2010 create a new WinForm application, name it as 'CS\_Working\_With\_Stream'. Design the Form as below:

ne



(Note: Set Names according to the image).

**Task 3:** Define the below objects at class level:

---

```
SqlConnection Conn;  
SqlCommand Cmd;  
string imageFilePath = null;
```

---

**Task 4:** Write the below code in the click event of the 'Display Image':

---

```
private void btnDisplayImage_Click(object sender, EventArgs e)  
{  
    try  
    {  
        OpenFileDialog dlg = new OpenFileDialog();  
        dlg.Filter = "JPEG File (*.jpg)|*.jpg|PNG File  
(*.*)|*.png";  
        DialogResult res = dlg.ShowDialog();  
        if (res == System.Windows.Forms.DialogResult.OK)
```

---

---

```

        {
            Bitmap b = new Bitmap(dlg.FileName);
            pictImgDisplay.Image = b;

            imagePath = dlg.FileName; //Get File Path
Here
        }

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

---

The above code will get the file and display into the PictureBox of name 'pictImgDisplay'.

**Task 5:** Write the below code in the click event of the Write Image to DB button:

---

```

private void btnWriteImage_Click(object sender, EventArgs e)
{
    try
    {
        byte[] file = null; //The byte Array where the file
bytes Will be Written

        //S1 : The FILE Info Class
        FileInfo fileInfo = new FileInfo(imageFilePath);

        //S2 : Get File Length
        long fileSize = fileInfo.Length;

        //S3 :Open the File in Read Mode

        FileStream Fs = new FileStream(imageFilePath,
FileMode.Open, FileAccess.Read);

        //S4: Initialize the Binary Reader to Read File into
Byte Array

        BinaryReader binReader = new BinaryReader(Fs);

        //S5 : Store all Bytes into Byte Array

```

---

---

```

        file = binReader.ReadBytes((int)fileSize);

        Fs.Close();
        Fs.Dispose();

        //S6: Database Login Starts Here
        Conn = new SqlConnection("Data Source=.;Initial
Catalog=Company;Integrated Security=SSPI");
        Conn.Open();
        Cmd = new SqlCommand();
        Cmd.Connection = Conn;
        Cmd.CommandText = "Insert into ImageTable Values
(@Image)";

        Cmd.Parameters.AddWithValue("@Image", (object)file);
        Cmd.ExecuteNonQuery();
        Conn.Close();
        Conn.Dispose();
        //S6 : Database logic Ends Here

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

---

The above code read the image into stream and byte by byte it is read and then after it is stored into DB.

**Task 6:** Write the below code in the click event of the Read Image from DB button:

---

```

private void btnReadImage_Click(object sender, EventArgs e)
{
    try
    {
        SqlConnection Conn = new SqlConnection("Data
Source=.;Initial Catalog=Company;Integrated Security=SSPI");
        SqlCommand Cmd = new SqlCommand();
        Conn.Open();
        Cmd.Connection = Conn;
        Cmd.CommandText = "Select * from ImageTable where
ImageId=@ImageId";
        Cmd.Parameters.AddWithValue("@ImageId",
Convert.ToInt32(txtimgid.Text));
        SqlDataReader Reader = Cmd.ExecuteReader();
        byte[] file = null;
        while (Reader.Read())

```

---

---

```
        {
            file = (byte[])Reader["Image"];
        }

        Reader.Close();

        //Convert Byte Array to Bitmap
        TypeConverter tc =
TypeDescriptor.GetConverter(typeof(Bitmap));
        Bitmap b = (Bitmap)tc.ConvertFrom(file);
        pictImgRead.Image = b;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

---

The above code read the Binary data from the database and then pass to the Bitmap using TypeConverter. This bitmap the passed to the Picture box.

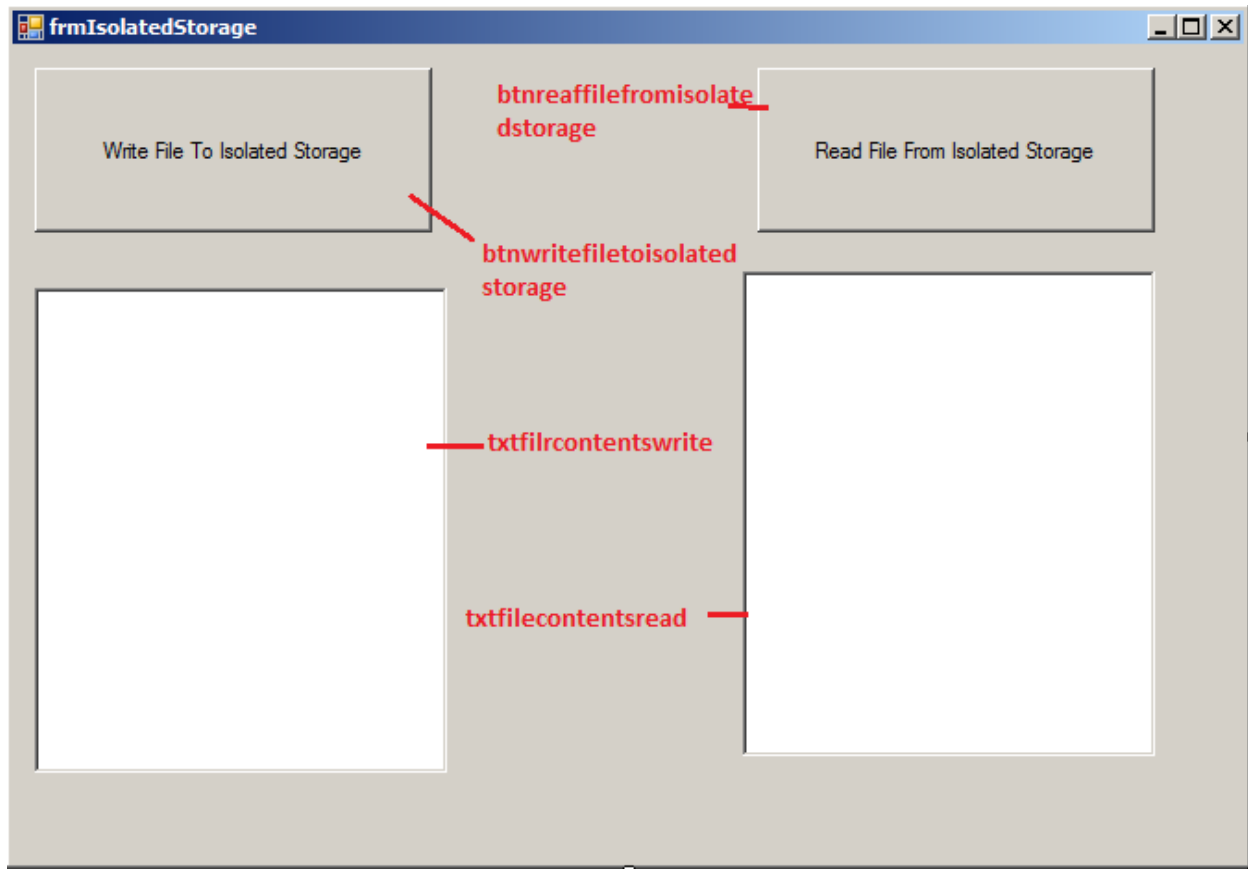
Run the application and Test it.

#### **Self Exercise Assignment:**

**Create a WinForm application which will store EmployeeInformation (EmpNo, EmpName, DeptName and Salary) into file. From the file Read Department wise employee.**

#### **Exercise 2: Working with Isolated Storage**

**Task 1:** Create a new Windows Application, name it as 'CS\_WinForm\_Isolated\_Storage'. Design the WinForm as below:



**Task 2:** Use below namespaces

```
using System.IO.IsolatedStorage;
using System.IO;
```

**Task 3:** Write the below code in click event of the Write File to Isolated Storage:

---

```
private void btnReadFromIsolatedStorage_Click(object sender, EventArgs
e)
{
    try
    {
        //S1 : Obtain the Isolated Storge
        using (IsolatedStorageFile file =
IsolatedStorageFile.GetMachineStoreForDomain())
        {
            //S2 : Define the Isolated Storage File SStream to
be created
            using (IsolatedStorageFileStream isoFs = new
IsolatedStorageFileStream("MyFile.txt", System.IO.FileMode.Open,
file))
```

---

---

```

        {
            using (StreamReader sr = new
StreamReader(isoFs))
            {
                txtFileContentsRead.Text= sr.ReadToEnd();
                sr.Close();
            }
            isoFs.Close();
        }
        file.Close();
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

---

Task 4: Write the below code in the click event of the Read File from Isolated Storage button:

---

```

private void btnReadFromIsolatedStorage_Click(object sender, EventArgs
e)
{
    try
    {
        //S1 : Obtain the Isolated Storge
        using (IsolatedStorageFile file =
IsolatedStorageFile.GetMachineStoreForDomain())
        {
            //S2 : Define the Isolated Storage File Stream to
be created
            using (IsolatedStorageFileStream isoFs = new
IsolatedStorageFileStream("MyFile.txt", System.IO.FileMode.Open,
file))
            {
                using (StreamReader sr = new
StreamReader(isoFs))
                {
                    txtFileContentsRead.Text= sr.ReadToEnd();
                    sr.Close();
                }
                isoFs.Close();
            }
            file.Close();
        }
    }
}

```

---

---

```
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message);  
    }  
}
```

---

Run and test the application.

**Self Exercise:**

Create an Employee Table in Sql Server Database with below columns:

EmpNo int Identity

EmpName varchar 50 not null

EmplImage image

Store Employee Data in the table.

Create a client application which will make a call to Employee to fetch the record. The Image of the Employee should be stored in IsolatedStorage storage if it is not already present in it. If the same record is called again then the image must be read from the isolated storage. User can delete the image from the isolated storage.